**Oracle® Fusion Middleware**

Administering the WebLogic Messaging Bridge for Oracle
WebLogic Server

12*c* (12.2.1.2.0)

**E77954-02**

December 2016

This document describes how to configure and manage a
WebLogic Messaging Bridge as a forwarding mechanism
between any two messaging products—thereby providing
interoperability between separate implementations of
WebLogic JMS, or between WebLogic JMS and another
messaging product.

**ORACLE**®

Oracle Fusion Middleware Administering the WebLogic Messaging Bridge for Oracle WebLogic Server, 12*c* (12.2.1.2.0)

E77954-02

# Contents

## 4    Interoperating with Different WebLogic Server Releases or Foreign Providers

## 5    FAQs: WebLogic Messaging Bridge

# Preface

This preface describes the document accessibility features and conventions used in this guide—.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

## Conventions

The following text conventions are used in this document:

| Convention | Meaning |
| --- | --- |
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

# 1

# Introduction and Roadmap

This chapter describes the contents and organization of this guide—*Administering the WebLogic Messaging Bridge for Oracle WebLogic Server*.

This chapter includes the following sections:

- Document Scope and Audience
- Guide to This Document
- Related Documentation
- Samples and Tutorials for the JMS Developer
- New and Changed Features in This Release

## 1.1 Document Scope and Audience

This document is a resource for system administrators who want to configure and manage a WebLogic Messaging Bridge as a forwarding mechanism between any two messaging products—thereby providing interoperability between separate implementations of WebLogic JMS, or between WebLogic JMS and another messaging product. It also contains information that is useful for business analysts and system architects who are evaluating WebLogic Server or considering the use of WebLogic Server JMS for a particular application.

It is assumed that the reader is familiar with programming in Java Platform, Enterprise Edition (Java EE) and JMS concepts. This document emphasizes the value-added features provided by WebLogic Server and key information about how to use WebLogic Server features and facilities to configure and manage a messaging bridge.

## 1.2 Guide to This Document

- This chapter, Introduction and Roadmap, describes the organization and scope of this guide, including new features and related documentation.

- Understanding the Messaging Bridge, describes basic WebLogic Messaging Bridge resources, such as resource adapters and destinations.

- Designing a Messaging Bridge, explains design options and other prerequisite considerations for configuring a WebLogic Messaging Bridge.

- Interoperating with Different WebLogic Server Releases or Foreign Providers, explains the interoperability guidelines that apply when using the messaging bridge to access JMS destinations on different releases of WebLogic Server and in other WebLogic Server domains.

- FAQs: WebLogic Messaging Bridge, provides answers to frequently asked questions (FAQs) for the WebLogic Messaging Bridge.

## 1.3 Related Documentation

For information on topics related to configuring and managing a messaging bridge, see the following documents:

- *Administering JMS Resources for Oracle WebLogic Server* is a guide to configuring and managing WebLogic JMS resources.

- *Developing JMS Applications for Oracle WebLogic Server* is a guide to developing WebLogic JMS applications.

- *Tuning Performance of Oracle WebLogic Server* provides information on how to monitor performance and tune the components in a WebLogic Server.

- *Developing Applications for Oracle WebLogic Server* is a guide to developing WebLogic Server applications.

- *Deploying Applications to Oracle WebLogic Server* is the primary source of information about deploying WebLogic Server applications.

- *Developing Resource Adapters for Oracle WebLogic Server* contains information on WebLogic resource adapters and the WebLogic Server implementation of the Java EE Connector Architecture.

## 1.4 Samples and Tutorials for the JMS Developer

In addition to this document, Oracle provides a variety of code samples and tutorials for JMS developers. The examples and tutorials illustrate WebLogic Server JMS in action, and provide practical instructions on how to perform key JMS development tasks.

Oracle recommends that you run some or all of the JMS examples before developing your own EJBs.

### 1.4.1 Avitek Medical Records Application (MedRec) and Tutorials

MedRec is an end-to-end sample Java EE application shipped with WebLogic Server that simulates an independent, centralized medical record management system. The MedRec application provides a framework for patients, doctors, and administrators to manage patient data using a variety of different clients.

MedRec demonstrates WebLogic Server and Java EE features, and highlights Oracle-recommended best practices. MedRec is optionally installed with the WebLogic Server installation. You can start MedRec from the `ORACLE_HOME`\user_projects \domains\medrec directory, where `ORACLE_HOME` is the directory you specified as the Oracle Home when you installed Oracle WebLogic Server. For more information on the WebLogic Server code examples, see Sample Applications and Code Examples in *Understanding Oracle WebLogic Server*.

MedRec includes a service tier comprised primarily of Enterprise Java Beans (EJBs) that work to process requests from web applications, web services, and workflow applications, and future client applications. The application includes message-driven, stateless session, stateful session, and entity EJBs.

### 1.4.2 JMS Examples in the WebLogic Server Distribution

WebLogic Server optionally installs API code examples in `WL_HOME\samples\server\examples\src\examples`, where `WL_HOME` is the top-level directory of your WebLogic Server installation. Start the examples server and obtain information about the samples and how to run them from the WebLogic Server Start menu.

### 1.4.3 Additional JMS Examples Available for Download

Additional API examples are available for download at the Oracle Technology Network at `http://www.oracle.com/technology/index.html`. These examples are distributed as .zip files that you unzip into an existing WebLogic Server samples directory structure. You build and run the downloadable examples in the same manner as you would an installed WebLogic Server example.

## 1.5 New and Changed Features in This Release

For a comprehensive listing of the new WebLogic Server features introduced in this release, see *What's New in Oracle WebLogic Server 12.2.1.2.0*.

# 2

# Understanding the Messaging Bridge

This chapter describes WebLogic Messaging Bridge components, concepts and functionality.

This chapter includes the following sections:

- What Is a Messaging Bridge?

- Messaging Bridge Components

- Configuring and Managing a Messaging Bridge

## 2.1 What Is a Messaging Bridge?

The WebLogic Messaging Bridge is a forwarding mechanism that provides interoperability between WebLogic JMS implementations, and between JMS and other messaging products. Use the Messaging Bridge to integrate your messaging applications between:

- Any two implementations of WebLogic JMS, including those from separate releases of WebLogic Server.

- WebLogic JMS implementations that reside in separate WebLogic domains.

- WebLogic JMS and a third-party JMS product (for example, MQSeries).

A messaging bridge instance forwards messages between a pair of bridge source and target destinations. These destinations are mapped to a pair of bridge source and target destinations. The messaging bridge reads messages from the source bridge destination and forwards those messages to the target bridge destination. For WebLogic JMS and third-party JMS products, a messaging bridge communicates with source and target destinations using the Java EE Connector Architecture (JCA) resource adapters provided with WebLogic Server. See Figure 2-1.

For non-JMS messaging products, a custom connector adapter must be obtained from a third-party OEM vendor or by contacting Oracle Professional Services in order to access non-JMS source or target destinations.

> **Note:**
>
> Messaging bridge configuration has been simplified since WebLogic Server 12.2.1. You need not explicitly configure and deploy bridge adapters. Instead, a messaging bridge now uses the automatically deployed internal adapters when the bridge is configured with a default adapter JNDI name (`eis.jms.WLSConnectionFactoryJNDINoTX` or `eis.jms.WLSConnectionFactoryJNDIXA`). These internal adapters also have a much larger connection capacity.
>
> Legacy adapters explicitly configured with the default adapter JNDI names still deploy, but are ignored by messaging bridges. The legacy adapter deployments can be removed in the configuration after upgrading to WebLogic Server 12.2.1.

*Figure 2-1    WebLogic Messaging Bridge*
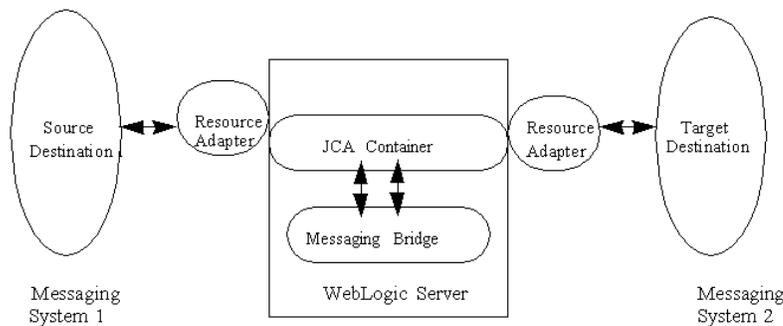


You designate source and target bridge destinations as either queues, topics, or distributed destinations, as described in *Developing JMS Applications for Oracle WebLogic Server*. Optionally, specify a quality of service (QOS), including message filters, transaction semantics, connection retry policies, and high availability options. Once you configure a messaging bridge you can perform management tasks easily from the WebLogic Server Administration Console, including suspending bridge traffic temporarily whenever necessary and monitoring the status of all your configured bridges.

# 2.2 Messaging Bridge Components

The following sections describe resources needed to use a messaging bridge:

- Resource Adapters
- Source and Target Bridge Destinations
- Messaging Bridge Instance

## 2.2.1 Resource Adapters

A messaging bridge uses JCA resource adapters to communicate with the configured source and target JMS destinations. Associate both the source and target JMS destinations with a *supported* resource adapter in order for the bridge to communicate with them. The JNDI name for the adapter is configured as part of the resource adapter's deployment descriptor. See Table 2-1.

> **Note:**
>
> Messaging bridge configuration has been simplified since WebLogic Server 12.2.1. You need not explicitly configure and deploy bridge adapters. Instead, a messaging bridge now uses the automatically deployed internal adapters when the bridge is configured with a default adapter JNDI name (`eis.jms.WLSConnectionFactoryJNDINoTX` or `eis.jms.WLSConnectionFactoryJNDIXA`). These internal adapters also have a much larger connection capacity.
>
> Legacy adapters explicitly configured with the default adapter JNDI names still deploy, but are ignored by messaging bridges. The legacy adapter deployments can be removed in the configuration after upgrading to WebLogic Server 12.2.1.

When a messaging bridge needs to use a custom configured adapter, explicit deployment of an adapter is required. In this case, the adapter must be configured with a non-default or custom adapter JNDI name, and the messaging bridge needs to be configured to reference the custom adapter JNDI name.

Resource adapters for different types of JMS destinations are provided in exploded format or in a `.rar` file. The exploded format gives you an easy way to modify resource adapter deployment descriptor parameters, such as the `max-capacity` of the connection factory that specifies the maximum number of connections available for bridge instances.

> **Note:**
>
> - Starting from WebLogic Server 12.2.1 release, the explored `.rar` format for bridge adapters is not available in the installation kit.
>
>   These are a couple of workarounds:
>
>   - Remove explicit adapter deployments and use the internally deployed adapters. The internal adapters set a higher `max-capacity` for the connection factory than the original `.rar` files.
>
>   - If you still prefer explicitly deploying an adapter, use one of the `.rar` files or generate the exploded format by using `jar -xf` on a `.rar` file.
>
> - Changing a deployment descriptor for a resource adapter using the exploded format does not update the descriptor packaged in the `.rar` file. See Setting the Number of Connection Factories.

The supported resource adapters are located in the *WL_HOME*`\server\lib` directory and are described in Table 2-1.

*Table 2-1 Messaging Bridge Resource Adapters and JNDI Names*

| Adapter | JNDI Name | Description |
|---------|-----------|-------------|
| jms-xa-adp | eis.jms.WLSConnectionFactoryJNDIXA | Provides transaction semantics the `XAResource`. Used when the required QOS is *Exactly-once*. This envelops a received message and sends it within a user transaction (XA/JTA). The following requirements apply to use of this resource adapter:<br>• Any WebLogic Server implementation being bridged must be release 7.0 or later.<br>• The source and target JMS connection factories must be configured to use the `XAConnectionFactory`.<br>**Note:** Before deploying this adapter, refer to the Configuring Interoperability for WebLogic Domains for specific transactional configuration requirements and guidelines. |
| jms-notran-adp | eis.jms.WLSConnectionFactoryJNDINoTX | Provides no transaction semantics. Used when the required QOS is *Atmost-once or Duplicate-okay*. If the requested QOS is *Atmost-once*, the resource adapter uses `AUTO_ACKNOWLEDGE` mode. If the requested QOS is *Duplicate-okay*, `CLIENT_ACKNOWLEDGE` is used.<br>**Note:** For more information about the acknowledge modes used in non-transacted sessions, see Understanding WebLogic JMS in *Developing JMS Applications for Oracle WebLogic Server*. |

## 2.2.2 Source and Target Bridge Destinations

A messaging bridge connects two actual destinations that are mapped to bridge destinations: a source destination *from which* messages are received, and a target destination *to which* messages are sent.

Depending on the messaging products that need to be bridged, there are two types of bridge destinations:

- JMS Bridge Destinations

- General Bridge Destinations

### 2.2.2.1 JMS Bridge Destinations

For JMS messaging products, whether it is a WebLogic JMS implementation or a third-party JMS provider, configure a JMS bridge destination instance for each actual source and target JMS destination being mapped to a messaging bridge. A JMS bridge destination instance defines a unique name for a bridge's source and target destinations within a WebLogic domain; the name of the adapter used to communicate with the specified destination; property information to pass to the adapter (such as Connection URL and Connection Factory JNDI Name), and, optionally, a user name and password. See Create JMS bridge destinations in *Oracle WebLogic Server Administration Console Online Help*. See:

- Interoperating with Different WebLogic Server Releases or Interoperating with Foreign Providers sections for specific configuration requirements and guidelines.

- When configuring third-party JMS provider bridge destination, use the Foreign JMS Server feature to configure multiple source or target destinations quickly. See Configuring Foreign Server Resources to Access Third-Party JMS Providers in *Administering JMS Resources for Oracle WebLogic Server*.

### 2.2.2.2 General Bridge Destinations

For non-JMS messaging products, you need to configure a general bridge destination instance for each actual source and target destination being mapped to a messaging bridge. A general bridge destination instance defines a unique name for the actual source and target general bridge destinations within the WebLogic domain; the name of the adapter used to communicate with the specified destination; a list of properties to pass to the adapter; and, optionally, a user name and password. See Create JMS bridge destinations in *Oracle WebLogic Server Administration Console Online Help*.

> **Note:**
>
> Although WebLogic JMS includes a "General Bridge Destination" framework for accessing non-JMS messaging products, WebLogic Server does not provide adapters for such products. Therefore, you must obtain a custom connector adapter from a third-party OEM vendor and consult their documentation for configuration instructions. Contact BEA Professional Services for information about obtaining a custom connector adapter.

## 2.2.3 Messaging Bridge Instance

A messaging bridge instance communicates with the configured source and target bridge destinations. For each mapping of a source destination to a target destination, whether it is another WebLogic JMS implementation or a third-party JMS provider, you must configure a messaging bridge instance. Each messaging bridge instance defines the source and target destination for the mapping, a message filtering selector, a QOS, transaction semantics, and various reconnection parameters. See Create Messaging Bridge Instances in *Oracle WebLogic Server Administration Console Online Help*.

Note that if the target destination is a non-JMS messaging product, you are still required to configure a messaging bridge instance.

# 2.3 Configuring and Managing a Messaging Bridge

The following sections provide information on how to use the WebLogic Server Administration Console to configure and manage a messaging bridge:

- Create a Messaging Bridge Instance

- Manage a Messaging Bridge Instance

## 2.3.1 Create a Messaging Bridge Instance

Creating a messaging bridge consists of the following tasks:

1. Create source and target bridge destinations.

2. Deploy a resource adapter.

3. Create a messaging bridge instance.

4. Configure any required high availability parameters. See Configuring Messaging Bridges for High Availability.

**5.** Target the messaging bridge.

The WebLogic Server Administration Console assists you in creating a messaging bridge by deploying an appropriate resource adapter and setting the values of some attributes. Consider changing messaging bridge settings to better suit your environment. See Create Messaging Bridge Instances in *Oracle WebLogic Server Administration Console Online Help*.

> **Note:**
>
> Targeting a messaging bridge to a mixed or dynamic cluster is not supported.

### 2.3.2 Manage a Messaging Bridge Instance

Typical tasks required to manage a messaging bridge using the WebLogic Server Administration Console include:

- Monitoring the status of all configured messaging bridges in your domain. See Monitor messaging bridge instances in the *Oracle WebLogic Server Administration Console Online Help*.

- Suspending and restarting an active messaging bridge. See Suspend and restart messaging bridges in the *Oracle WebLogic Server Administration Console Online Help*.

- Configuring the default execute thread pool size for your messaging bridges. See Configure messaging bridge execute thread pool size in the *Oracle WebLogic Server Administration Console Online Help*.

- Deploying a resource adapter. See Deploy resource adapters in the *Oracle WebLogic Server Administration Console Online Help*.

- Creating a trusted security relationship. See Configuring Secure Inter-Domain and Intra-Domain Transaction Communication in *Developing JTA Applications for Oracle WebLogic Server*.

## 2.4 Configuring Messaging Bridges for High Availability

Similar to a persistent store, a message bridge can be targeted to a cluster and configured for high availability and dynamic scalability. When the distribution policy property of the messaging bridge is set to **Distributed**, then one instance of the bridge will be created and started on each cluster member. If the distribution policy is set to **Singleton**, then only a single instance of the bridge will be started in the cluster. See Defining the Distribution Policy for JMS Services in *Administering JMS Resources for Oracle WebLogic Server*.

Bridges require the migration policy set to **On-Failure** or **Off**. A validation error will be logged when the users try to configure the migration policy as **Always**. Bridges already have the built-in mechanism to retry failed operations. Therefore, bridges ignore the restart-in-place settings. For more information, see Defining the Migration Policy for JMS Services in *Administering JMS Resources for Oracle WebLogic Server*.

# 3

# Designing a Messaging Bridge

This chapter describes how to design, configure, and tune a Messaging Bridge.

This chapter includes the following sections:

- When to use a Messaging Bridge

- When to Avoid using a Messaging Bridge

- Selecting a Quality-of-Service (QOS) Level

- Messaging Persistence

- Message Ordering

- Setting the Number of Connection Factories

- Preserving Message Properties

- Using the JMSXUserID Property with a Messaging Bridge

- Using Distributed Destinations as Source and Target Destinations

- Tuning a WebLogic Messaging Bridge

## 3.1 When to use a Messaging Bridge

The following sections provide information on when to use a messaging bridge:

- Store and Forward Messaging

- Replicating a Topic

### 3.1.1 Store and Forward Messaging

A messaging bridge provides high availability for remote destinations. Store and forward messaging enables a local client to produce to a local destination and have those messages automatically forwarded to the remote destination when it is available. This allows a local client to continue to produce messages when a remote destination is not available. See Messaging Persistence.

Use the WebLogic Messaging Bridge to provide an administrative solution to store and forward messages between:

- Any two implementations of WebLogic JMS, including those from separate releases of WebLogic Server.

- WebLogic JMS implementations that reside in separate WebLogic domains.

- WebLogic JMS with a third-party JMS product (for example, MQSeries).

WebLogic JMS with non-JMS messaging products (only by using specialized connector adapters that are not provided with WebLogic Server)

### 3.1.2 Replicating a Topic

A messaging bridge can replicate a topic, similar to using the distributed topics feature available in WebLogic Server releases, consequently improving scalability and high availability in some scenarios. (For information about using distributed topics, see Using Distributed Destinations in *Developing JMS Applications for Oracle WebLogic Server*.) Topic replication is accomplished by configuring the bridge to subscribe to one topic and forward the topic's messages to another topic, in essence creating two topics with the same message stream. See Create messaging bridge instances in *Oracle WebLogic Server Administration Console Online Help*.

## 3.2 When to Avoid using a Messaging Bridge

The following sections provide information on when to avoid using messaging bridge:

- Receiving messages from a remote destination—Use a message driven EJB or implement a client consumer directly.

- Sending messages to a local destination—Send directly to the local destination.

- Environment with low tolerance for message latency. Messaging Bridges increase latency and may lower throughput. Messaging bridges increase latency for messages as they introduce an extra destination in the message path and may lower throughput because they forward messages using a single thread.

- Forward messages between WebLogic domains—Use WebLogic Store-and-Forward.

Table 3-1 summarizes information on when to use WebLogic Messaging Bridge or other forwarding technologies:

**Table 3-1    Comparing Message Forwarding Technologies**

| Feature | Messaging Bridge | Message Driven Beans | WebLogic Store-and-Forward |
|---|---|---|---|
| Implementation mechanism | Administrative | Programmatic | Administrative |
| Support for foreign and legacy providers | Yes | No | No, use to forward messages |

## 3.3 Selecting a Quality-of-Service (QOS) Level

The WebLogic Messaging Bridge supports three different QOS levels:

- Exactly-once—The highest QOS guarantees that a message is sent to the remote endpoint once and only once. With Exactly-once, messages survive server crashes and network down time, while guaranteeing one occurrence of each message at the endpoint.

- At-least-once—Guarantees that a message is sent to the remote endpoint, but with the possibility of duplicates. With `At-least-once`, multiple copies of a message

may appear on the remote endpoint because of network failures or server crashes that occur when the message is in transit.

- At-most-once—The lowest QOS guarantees that each message is sent to the remote endpoint only once, if at all. It does not guarantee that a message is sent to the endpoint. With At-most-once, messages may get lost because of network failures or server crashes. No duplicate messages reach the endpoint.

In some instances, the target destination may not be able to provide the quality of service configured for the bridge. In these cases, configure the bridge instance to allow the quality of service to be degraded by setting the `QOSDegradationAllowed` flag. See Create messaging bridge instances in *Oracle WebLogic Server Administration Console Online Help*.

## 3.4 Messaging Persistence

Store-and-forward messaging enables a local JMS client to produce messages to a local destination and have those messages automatically forwarded to a remote destination when it is available. The bridge forwards these messages to the target destination when it is restarted. A messaging bridge stores and forwards messages to a target destination under the following conditions:

- The source destination is a queue.

- The source destination is a topic *and* the Durability Enabled attribute is set. This creates a durable subscription. For more information configuring durable topic subscribers, see Setting Up Durable Subscriptions in *Developing JMS Applications for Oracle WebLogic Server*.

## 3.5 Message Ordering

If an application message is in a transaction, saving the message in the persistent store must be part of the user transaction to preserve exactly-once semantics. In particular, the message is removed from the persistent store as part of the transaction rollback if the application decides to rollback the transaction. However, forwarding is *not* part of the application transaction. The sending agent does not forward a transactional message until the transaction commits. Within a transaction, message ordering is preserved based on when the messages are sent.

To ensure message ordering, configure a message unit-of-order. See Using Message Unit-of-Order in *Developing JMS Applications for Oracle WebLogic Server*.

## 3.6 Setting the Number of Connection Factories

Consider modifying the capacity of the connection factory associated with each resource adaptor by adjusting the `max-capacity` attribute in the `weblogic-ra.xml` descriptor file. In general, the value of the `max-capacity` attribute should be at least two times the number of bridge instances. For example, if your environment has up to ten message bridge instances targeted, a `max-capacity` attribute setting of 20 in the default configuration is adequate. But if you increase the number of bridge instances to 15, increase the `max-capacity` attribute to 30.

> **Note:**
>
> Do not change Message Bridge adapter `initial-capacity` value. Oracle recommends that the `initial-capacity` value be set to zero (the default) as bridge instances may fail to function with `ResourceAllocation` errors.

Use the following steps to modify the `weblogic-ra.xml` descriptor file:

1. Using the editor of your choice, update the attribute with the desired value. See Example 3-1.

2. Deploy the updated adapter.

3. Stop and restart any bridge instance that requires the new values.

### Example 3-1    Example weblogic-ra.xml Descriptor File

```
<weblogic-connection-factory-dd>

    <connection-factory-name>WLSJMSConnectionFactoryLocal</connection-factory-name>
    <jndi-name>eis/jms/WLSConnectionFactoryJNDILocal</jndi-name>
        <pool-params>
            <initial-capacity>0</initial-capacity>
            <max-capacity>20</max-capacity>
        </pool-params>

</weblogic-connection-factory-dd>
```

## 3.7 Preserving Message Properties

Set `PreserveMsgProperty` to preserve message properties in a message header when a message is forwarded by a bridge instance. In previous releases, message properties are inherited from the `Default Delivery Mode` attribute on the connection factory used when a message is forwarded to its target destination. If the `Default Delivery Mode` is persistent, a non-persistent message is forwarded as a persistent message resulting in a significant performance loss.

When `PreserveMsgProperty` is enabled, an incoming non-persistent message is forwarded by the bridge to the target destination as a non-persistent message and an incoming persistent message is forwarded to the target destination as a persistent message. See Configure messaging bridge instances in *Oracle WebLogic Server Administration Console Online Help*.

The behavior of a messaging bridge instance is determined according to the following guidelines:

- The `PreserveMsgProperty` is not enabled. This setting provides the same forwarding behavior as previous releases.

- The default value of `PreserveMsgProperty` when configuring a messaging bridge instance is not enabled.

- The `PreserveMsgProperty` is enabled. Message properties are preserved as described in Table 3-2:

*Table 3-2    Message Properties Preserved at Target Destination by WebLogic Server Release*

| Property | WebLogic Server 9.0 and Higher | Prior to WebLogic Server 9.0 | Foreign JMS Servers |
|---|---|---|---|
| Message ID | Yes | No | No |
| Timestamp | Yes | No | No |
| User ID | Yes | No | No |
| Delivery Mode | Yes | Yes | Yes |
| Priority | Yes | Yes | Yes |
| Expiration Time | Yes | Yes | Yes |
| Redelivery Limit | Yes | No | No |
| Unit-of-Order name | Yes | No | No |

## 3.8 Using the JMSXUserID Property with a Messaging Bridge

The messaging bridge does not disclose a message's JMSXUserID across messaging bridge boundaries. A JMSXUserID is a system generated property that identifies the user sending the message, see the JMS Specification, published by Oracle, at http://www.oracle.com/technetwork/java/jms/index.html.

## 3.9 Using Distributed Destinations as Source and Target Destinations

A messaging bridge can send to and receive from distributed destinations. Oracle recommends the following configurations:

- If the source is a distributed destination, the bridge is pinned to one of the members when it connects to the destination. It stays connected to that member until an event occurs that breaks the connection. On reconnection, the bridge uses the next available member. Once a bridge is connected, it does not receive messages from other members of the distributed destination. It is a best practice to configure one bridge for each member of a distributed destination using the member's JNDI Name.

- If the target is a distributed destination, the best practice is to send to the distributed destination using the distributed destination's JNDI Name and disable server affinity. This allows the distributed destination to load balance incoming messages. See Load Balancing for JMS in *Administering Clusters for Oracle WebLogic Server*.

## 3.10 Tuning a WebLogic Messaging Bridge

The main objective when tuning a messaging bridge is to improve overall messaging performance. Raw speed, though important, is only one of several performance-related factors. Other performance factors include reliability, scalability, manageability, monitoring, user transactions, message-driven bean support, and integration with an application server. See Tuning WebLogic Message Bridge in *Tuning Performance of Oracle WebLogic Server*.

# 4

# Interoperating with Different WebLogic Server Releases or Foreign Providers

This chapter provides interoperability guidelines for using the WebLogic Messaging Bridge to access JMS destinations on different releases of WebLogic Server or when accessing foreign providers.

This chapter includes the following sections:

- Interoperating with Different WebLogic Server Releases
- Interoperating with Foreign Providers

## 4.1 Interoperating with Different WebLogic Server Releases

The following interoperability guidelines apply when using the messaging bridge to access JMS destinations on different releases of WebLogic Server and in other WebLogic Server domains.

- Supported Configurations and Compatibility
- Naming Guidelines for WebLogic Servers and Domains
- Configuring Interoperability for WebLogic Domains

---

**Note:**

When the messaging bridge is used to communicate between two domains running different releases of WebLogic Server, Oracle recommends that the messaging bridge be configured to run on the domain using the latest release of WebLogic Server.

---

### 4.1.1 Supported Configurations and Compatibility

For the most current information on supported configurations and compatibility between current version of WebLogic Server and previous releases, see Supported Configurations and WebLogic Server Compatibility in *What's New in Oracle WebLogic Server 12.2.1.2.0*.

### 4.1.2 Naming Guidelines for WebLogic Servers and Domains

Unique naming rules apply to all WebLogic Server deployments if more than one domain is involved. For more information, see Integration and Multi-Domain Best Practices in *Administering JMS Resources for Oracle WebLogic Server*.

### 4.1.3 Configuring Interoperability for WebLogic Domains

Configuring interoperability between domains depends on the QOS:

- Configuring Interoperability for Exactly-once QOS

- Configuring At-least-once or At-most-once QOS

#### 4.1.3.1 Configuring Interoperability for Exactly-once QOS

Use the following guidelines to configure interoperability to handle transactional messages (using the Exactly-once QOS) between two WebLogic domains

- You must correctly configure either Cross Domain Security or Security Interop Mode for all participating domains.

  Keep all the domains used by your process symmetric with respect to Cross Domain Security configuration and Security Interop Mode. Because both settings are set at the domain level, it is possible for a domain to be in a mixed mode, meaning the domain has both Cross Domain Security and Security Interop Mode set. For more information, see:

  - Configuring Secure Inter-Domain and Intra-Domain Transaction Communication in *Developing JTA Applications for Oracle WebLogic Server*

  - Changing the Asynchronous Mode Enabled Attribute in *Tuning Performance of Oracle WebLogic Server*

- When interoperating between two WebLogic domains, see Naming Guidelines for WebLogic Servers and Domains.

- Ensure that the XA connection factory is enabled for the domains by selecting the XAConnection Factory Enabled check box. See Configure connection factory transaction parameters in *Oracle WebLogic Server Administration Console Online Help*.

- Deploy the transaction resource adapter, jms-xa-adp.rar, on the domain where the messaging bridge is running, as described in Deploy resource adaptors in *Oracle WebLogic Server Administration Console Online Help*.

- When configuring the JMS bridge destinations, as described in Create JMS bridge destinations in *Oracle WebLogic Server Administration Console Online Help*, do the following for both the source and target destinations:

  - In the Adapter JNDI Name field, identify the transaction adapter's JNDI name, eis.jms.WLSConnectionFactoryJNDIXA.

  - Do not enter anything in the Adapter Classpath field.

- Select a Quality Of Service of *Exactly-once*, as described in Configure messaging bridge instances in *Oracle WebLogic Server Administration Console Online Help*.

#### 4.1.3.2 Configuring At-least-once or At-most-once QOS

There are no special security configuration requirements for the bridge to interoperate between two WebLogic domains. However, if you want to provide more secure communication between domains, you can configure Cross Domain Security.

> **Note:**
>
> Keep all the domains used by your process symmetric with respect to Cross Domain Security configuration—all domains use Cross Domain Security (or are on the appropriate exception lists) or none of the domains have Cross Domain Security configured. For more information, see:
>
> - Configuring Cross-Domain Security in *Administering Security for Oracle WebLogic Server*
>
> - Changing the Asynchronous Mode Enabled Attribute in *Tuning Performance of Oracle WebLogic Server*

## 4.2 Interoperating with Foreign Providers

When configuring a messaging bridge involves interoperability with a third-party messaging provider, you must configure the following:

- Before starting WebLogic Server:

  - Supply the provider's `CLASSPATH` in the WebLogic Server `CLASSPATH`.

  - Include the `PATH` of any native code required by the provider's client-side libraries in the WebLogic Server system `PATH`. (This variable may vary depending on your operating system.)

- In the `JMSBridgeDestination` instance for the third-party messaging product being bridged, provide *vendor-specific* information in the following attributes:

  - Connection URL

  - Initial Context Factory

  - Connection Factory JNDI Name

  - Destination JNDI Name

  > **Note:**
  >
  > The messaging bridge cannot provide the "Exactly-once" quality of service when the source and target bridge destinations are located on the same resource manager (that is, when the bridge is forwarding a global transaction that is using the XA resource of the resource manager). For example, when using MQ Series, it is not possible to use the same Queue Manager for the source and target bridge destinations.

For more information on configuring the remaining attributes for a JMS Bridge Destination, see Create JMS Bridge destinations in *Oracle WebLogic Server Administration Console Online Help*.

# 5

# FAQs: WebLogic Messaging Bridge

This chapter provides answers to frequently asked questions (FAQs) for the WebLogic Messaging Bridge.

This chapter includes the following sections:

- Why did the messaging bridge fail to connect to the source bridge destination?

- Why are some of my messages not being forwarded?

- Can the messaging bridge handle two-phase or global transactions between separate WebLogic Server domains or between different releases?

- I configured the messaging bridge to use the Exactly-once quality of service for two-phase transactions. So why am I getting a "quality of service is unreachable" error?

- Can a messaging bridge to automatically downgrade the quality of service if the Exactly-once service is not available on either the source or target bridge destination?

- I deployed the transactional jms-xa-adp.rar resource adapter on domain where the message bridge is running, but I still get a "failed to find bridge adapter" message?

- When configuring a source or target messaging bridge destination, do I need to set the Adapter Classpath field?

- How do I enable debugging for the messaging bridge?

- What do the messaging bridge monitoring states indicate on the Monitor Messaging Bridge console page?

- Is there another way to monitor the messaging bridge without using the Administration Console?

- Can the messaging bridge use distributed destinations as source and target destinations?

- Can I use the Message Bridge in a Mixed or Dynamic Cluster?

## 5.1 Why did the messaging bridge fail to connect to the source bridge destination?

Either an error occurred when configuring the source bridge destination parameters, or the actual source destination is not running and cannot communicate with the messaging bridge.

- Verify whether the bridge's source destination is correctly configured, by making sure that the following fields on the JMS Bridge Destination Æ Configuration console page have been properly completed:

  - Connection URL—this must be the URL of the JNDI provider used to look up the connection factory and actual destination.

  - Destination JNDI Name—this must be the JNDI name of the actual destination mapped to the source bridge destination.

  - Connection Factory JNDI Name—this must be the connection factory used to create a connection for the actual destination mapped to the source bridge destination.

  - User Name/Password—ensure that this user ID has permission to access the actual source destination.

- Verify that the actual source queue or topic destination mapped to the source bridge destination is running and healthy, as follows:

  - Is the WebLogic Server instance hosting the source destination running?

  - Is the JMS server hosting the source destination correctly deployed?

  > **Note:**
  >
  > This troubleshooting scenario for correcting a source bridge destination connection failure also applies to target bridge destinations.

## 5.2 Why are some of my messages not being forwarded?

Normally, a messaging bridge should forward all messages. If some messages are not being forwarded, here are some possible reasons:

- Some messages may have an expiration time, in which case either the JMS provider for the source or target destination expires the message.

- If you configured the bridge source destination to specify a selector filter, only the filtered messages are forwarded.

- A bridge does not directly provide an option to automatically move messages to an error destination, or to automatically delete messages, after a limited number of forward attempts. However, a JMS provider may provide such an option, which could, in turn, effect any messages on the bridge source destination. If a redelivery limit option is enabled on the JMS provider that hosts the bridge source destination, consider reconfiguring the provider to prevent the bridge automatic retry mechanism from causing messages to exceed the redelivery limit.

## 5.3 Can the messaging bridge handle two-phase or global transactions between separate WebLogic Server domains or between different releases?

Yes, see Interoperating with Different WebLogic Server Releases.

## 5.4 I configured the messaging bridge to use the Exactly-once quality of service for two-phase transactions. So why am I getting a "quality of service is unreachable" error?

There are some additional configuration requirements for the messaging bridge to handle transactions between WebLogic domains:

- The supported adapters are located in the `WL_HOME\server\lib` directory. For the `Exactly-once` QOS, the transaction adapter, `jms-xa-adp.rar`, must be deployed in the domain where the bridge is running, through the select Deployments > Connector node on the console.

- This `jms-xa-adp.rar` adapter must also be identified in the Adapter JNDI Name attribute as `eis.jms.WLSConnectionFactoryJNDIXA` on the JMS Bridge Destination > Configuration tab for both the source and target bridge destinations.

- For WebLogic JMS, verify that you are using the transactional `XAConnectionFactory` for the queue or topic destinations mapped to both the source and target bridge destinations. Verify by reviewing the **JMS > Connection Factory > Configuration > Transactions** console tab or in the configuration file (config.xml):

  ```
  UserTransactionsEnabled=true
  XAConnectionFactory=true
  ```

- For third-party JMS vendors, verify that you are using a transactional connection factory for the destinations mapped to the source and target bridge destinations.

For more information about using the Exactly-once QOS when interoperating between releases, see Interoperating with Different WebLogic Server Releases.

## 5.5 Can a messaging bridge to automatically downgrade the quality of service if the Exactly-once service is not available on either the source or target bridge destination?

Yes, just ensure to select the `QOS Degradation Allowed` check box on the **Messaging Bridge** > **Configuration** > **General** WebLogic Server Administration Console page.

## 5.6 I deployed the transactional jms-xa-adp.rar resource adapter on domain where the message bridge is running, but I still get a "failed to find bridge adapter" message?

Associate both the source and target bridge destinations with the appropriate.rar adapters in order for the bridge to communicate with them. For the `jms-xa-adp.rar` transaction adapter, it must be identified in the Adapter JNDI Name attribute as eis.jms.WLSConnectionFactoryJNDIXA on the JMS Bridge Destination Æ Configuration tab for both the source and target bridge destinations.

> **Note:**
>
> The "failed to find bridge adapter" message does not necessarily indicate a problem if it only occurs once. However, if it occurs repeatedly, you should check the adapter deployment and the adapter JNDI name used in the source and target bridge destinations.

For more information about the bridge resource adapters, see Resource Adapters.

## 5.7 When configuring a source or target messaging bridge destination, do I need to set the Adapter Classpath field?

Leave the Adapter Classpath field blank when connecting to source and target destinations that are both running in WebLogic Server instances. When connecting to a third-party JMS provider, the bridge destination must supply the provider's CLASSPATH in the WebLogic Server CLASSPATH.

## 5.8 How do I enable debugging for the messaging bridge?

You can enable debugging for the messaging bridge using either of the followings methods:

- Add the following lines to your WebLogic start script (before the weblogic.Server line):

```
-Dweblogic.debug.DebugMessagingBridgeStartup=true
-Dweblogic.debug.DebugMessagingBridgeRuntime=true
```

- Add the following statements to the ServerDebug entry in your configuration file (config.xml) for the server that the messaging bridge is running on:

```
DebugMessagingBridgeStartup="true"
DebugMessagingBridgeRuntime="true"
```

Once debugging is enabled for the messaging bridge, the debugging messages are sent to the server log by default. However, if you want them to appear in the WebLogic Server Administration Console, add "DumpToConsole" to the statements show above. For example:

```
-Dweblogic.debug.DebugMessagingBridgeStartupDumpToConsole=true
```

## 5.9 What do the messaging bridge monitoring states indicate on the Monitor Messaging Bridge console page?

When monitoring a messaging bridge's state, use Table 5-1 to determine a course of action, if necessary. For more information, see Manage a Messaging Bridge Instance.

**Table 5-1    Messaging Bridge Monitoring States**

| Description | Action |
| --- | --- |
| WARN: Failed to find the source adapter | Check if the adapter is deployed or the JNDI name in the source JMSBridgeDestination instance is correct. |

*Table 5-1    (Cont.) Messaging Bridge Monitoring States*

| Description | Action |
| --- | --- |
| WARN: Failed to find the target adapter | Check if the adapter is deployed or the JNDI name in the target JMSBridgeDestination instance is correct. |
| Found both of the adapters and making connections | NA |
| WARN: Stopped by the administrator | NA |
| WARN: Failed to look up the source adapter | Check if the adapter is deployed or the JNDI name in the source JMSBridgeDestination instance is correct. |
| WARN: Failed to look up the target adapter | Check if the adapter is deployed or the JNDI name in the target JMSBridgeDestination instance is correct. |
| Found two adapters and about to make connections | NA |
| WARN: Failed to connect to the source | • Check all the parameters configured for the source bridge destination.<br>• Check if the source server is running and whether the actual destination is active. |
| Connected to the source | NA |
| WARN: Failed to connect to the target | • Check all the parameters configured for the target bridge destination.<br>• Check if the target server is running and whether the actual destination is active. |
| Connected to the target | NA |
| Forwarding messages | NA |
| WARN: Failed to connect and will reconnect later | Check if the source and target bridge destinations are running and healthy. |

## 5.10 Can the messaging bridge use distributed destinations as source and target destinations?

Yes, the messaging bridge can send to and receive from distributed destinations. Oracle recommends the following configurations:

• If the source is distributed destination, the bridge is pinned to one of the members when it connects to the destination. It stays connected only to that member until it reconnects. The bridge does not receive messages from the other members of the

distributed destination. Therefore, the best practice is to configure one bridge for each member of a distributed destinations using the member's `JNDIName`.

- If the target is a distributed destination, the best practice is to send to the distributed destination using the distributed destination's `JNDIName` and disable server affinity. This allows the distributed destination to load balance incoming messages.

## 5.11 Is there another way to monitor the messaging bridge without using the Administration Console?

Yes, there is a run-time MBean (`MessagingBridgeRuntimeMBean`) for each bridge instance. WebLogic Server run-time MBeans provide a snapshot of information about domain resources. When a particular resource in the domain (such as a messaging bridge) is instantiated, an MBean instance is created which collects information about that resource.

The `MessagingBridgeRuntimeMBean` has a `getState()` method that currently returns a `String` ("Active" or "Inactive") and a `getDescription()` method, which returns a `String` with more detailed information. The name of a bridge runtime MBean consists of the WebLogic Server instance name and the bridge name. If a bridge named *mybridge*, runs on WebLogic Server instance named *myserver*, the bridge runtime MBean is named *myserver.bridge.mybridge*.

For more information, see:

- Introduction and Roadmap in *Developing Custom Management Utilities Using JMX for Oracle WebLogic Server*

- Using the WebLogic Scripting Tool in *Understanding the WebLogic Scripting Tool*

## 5.12 Can I use the Message Bridge in a Mixed or Dynamic Cluster?

No. Targeting a message bridge in a mixed or dynamic cluster is not supported.